


[What is this page?](#)

voicexml re-entrant

Search

powered by
Google
☒ Search WWW ☐ Search Centipedia.com

Web

Results 1 - 10 of about 34 for voicexml re-entrant. (0.55 seconds)

powered by
Google

A Re-entrant Reusable Dialog Component Framework for Context ...

This is achieved by requiring that objects and subdialogs be **re-entrant**: they ... When used in the context of **VoiceXML**, this proposal requires extensions of ...
csdl.computer.org/comp/proceedings/icme/2001/1198/00/11980187abs.htm - [Similar pages](#)

United States Patent Application: 0020198719

The speech application server of claim 13, wherein a reusable **VoiceXML** dialog component is **re-entrant**, allowing the reusable **VoiceXML** dialog components to ...
appft1.uspto.gov/.../Gergic&RS=IN/Gergic - Dec 1, 2005 - [Similar pages](#)

United States Patent Application: 0020052747

A voice browser comprising: a **reentrant** interpreter maintaining separate contexts of ... Examples of voice markup languages include VoxML, **VoiceXML**, etc. ...
appft1.uspto.gov/.../20020052747&RS=DN/20020052747 - [Similar pages](#)

terry chay

"Effects of Channel Blockers on **Reentrant** Arrhythmias: Model Study. ... Implemented wireless presentation (AvantGo, PQA, WAP, **VoiceXML**, XSL-T). ...
homepage.mac.com/tychay/Resume.html - [Similar pages](#)

Topics in Internet Programming

reentrant CGI programs; separation of service logic from user interface ... from 800 toll-free numbers to **Voice-XML**", Ken Rehor, Software Production ...
www1.math.luc.edu/~tuckey/ - [Similar pages](#)

Mitch Ratcliffe's Travels

Mike M: TellMe Studio – anyone can build a **voice XML** system and deliver it over the phone. ... **Re-entrant** conversations. Persistent use throughout life. ...
radio.weblogs.com/0116895/2002/12/10.html - [Similar pages](#)

Interface to Mikropuhe for writing files. // TAB=3 spaces ...

... SAPI5-tags are allowed //MPINT_TAGS_VXML = 8 // **VoiceXML**/SSML tags, ... Functions are not **re-entrant** // 3. You may have many channels each running their ...
www.mikropuhe.com/ohjeet/MPWRFILE.H - [Similar pages](#)

gmane.comp.jakarta.taglibs.user

... was written on the concept that servlet/jsp api is not **re-entrant**, ... the RDC to work as a subdialog (which includes a **VoiceXML** <return> element). ...
blog.gmane.org/gmane.comp.jakarta.taglibs.user - [Similar pages](#)

DBLP: Stéphane H. Maes

8, EE, Stéphane H. Maes: A **VoiceXML** Framework for Reusable Dialog Components. ... 5, EE, Stéphane H. Maes: A **Re-entrant** Reusable Dialog Component Framework ...

www.informatik.uni-trier.de/~ley/db/indices/a-tree/m/Maes:St=eacute=phane_H=.html - [Similar pages](#)

[0.104 12 Nov 2005 access-types source data date change \(no ...](#)
... srgs srgs+xml ssml+xml **voicexml**+xml xhtml-voice+xml Received field "via" ...
functions uses flex 2.5.11 or later for **reentrant** lexical analyzer fixed ...
users.erols.com/blilly/mparse/CHANGES - [Similar pages](#)

Result Page: [1](#) [2](#) [3](#) [Next](#)

[Search within results](#) | [Search Tips](#)

©2005 Google


[Subscribe](#) (Full Service) [Register](#) (Limited Service, Free) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

VoiceXML for Web-based distributed conversational applications

 Full text [Html](#) (22 KB), [Pdf](#) (80 KB)

Source [Communications of the ACM](#) [archive](#)
 Volume 43, Issue 9 (September 2000) [table of contents](#)
 Pages: 53 - 57
 Year of Publication: 2000
 ISSN:0001-0782

Author [Bruce Lucas](#) IBM T.J. Watson Research Center, Yorktown Heights, NY

Publisher ACM Press New York, NY, USA

Additional Information: [citations](#) [index terms](#) [collaborative colleagues](#) [peer to peer](#)

Tools and Actions: [Discussions](#) [Find similar Articles](#) [Review this Article](#)
[Save this Article to a Binder](#) Display Formats: [BibTex](#) [EndNote](#) [ACM Ref](#)

DOI Bookmark: Use this link to bookmark this Article: <http://doi.acm.org/10.1145/348941.348985>
[What is a DOI?](#)

↑ CITINGS 2

[Juliana Freire](#), [Bharat Kumar](#), [Daniel Lieuwen](#), [WebViews: accessing personalized web content and services](#), Proceedings of the tenth international conference on World Wide Web, p.576-586, May 01-05, 2001, Hong Kong, Hong Kong

[Clare-Marie Karat](#), [John Vergo](#), [David Nahamoo](#), [Conversational interface technologies, The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications](#), Lawrence Erlbaum Associates, Inc., Mahwah, NJ, 2002

↑ INDEX TERMS

Primary Classification:

I. [Computing Methodologies](#)

↳ I.2 [ARTIFICIAL INTELLIGENCE](#)

↳ I.2.1 [Applications and Expert Systems](#)

↳ **Subjects:** [Natural language interfaces](#)

Additional Classification:

I. [Computing Methodologies](#)

↳ I.7 [DOCUMENT AND TEXT PROCESSING](#)

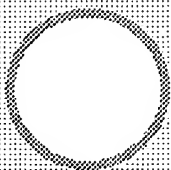
↳ I.7.2 [Document Preparation](#)

↳ **Nouns:** [XML](#)

Bruce Lucas

VOICEXML *for* WEB-BASED DISTRIBUTED CONVERSATIONAL APPLICATIONS

VoiceXML replaces the familiar HTML interpreter (Web browser) with a VoiceXML interpreter and the mouse and keyboard with the human voice.



UNTIL RECENTLY, THE WEB DELIVERED INFORMATION AND SERVICES exclusively through visual interfaces on computers with displays, keyboards, and pointing devices. The Web revolution largely bypassed the huge market for information and services represented by the worldwide installed base of telephones, for which voice input and audio output provide the sole means of interaction.

Development of speech services has been hindered by a lack of easy-to-use standard tools for managing the dialogue between user and service. Interactive voice-response systems are characterized by expensive, closed application-development environments. Lack of tools inhibits portability of applications and limits the availability of skilled application developers. Consequently, voice appli-

cations are costly to develop and deploy, so voice access is limited to only those services for which the business case is most compelling for voice access.

Here, I offer an introduction to VoiceXML, an emerging standard XML-based markup language for distributed Web-based voice services, much as HTML is a language for distributed visual services. VoiceXML brings the power of Web development

and content delivery to voice-response applications, freeing developers from low-level programming and resource management. It also enables integration of voice services with data services, using the familiar client/server paradigm and leveraging the skills of Web developers to speed application development for this new medium.

VoiceXML 1.0 was developed by the VoiceXML Forum (see www.voicexml.org), which released it in March 2000, and was accepted by the World Wide Web Consortium (W3C) two months later as the basis for developing a W3C dialogue markup language (see www.w3.org/Voice/). The initial version of the language included robust support for basic state-based dialogue capabilities, using a design with simple form-based natural language capabilities that leaves room to grow as the technology evolves. (You can find the VoiceXML specification at www.voicexml.org and a free implementation for personal use at www.alphaworks.ibm.com/tech/voic-eserversdk.)

Basic Spoken Dialogue

The basic spoken dialogue capabilities of VoiceXML are illustrated by the following VoiceXML document containing a menu and a form:

```
<?xml version="1.0"?>
<vxml version="1.0">

  <menu>
    <prompt>Say one of: <enumerate/></prompt>
    <choice next="http://www.sports.example/sports.vxml">
      Sports scores
    </choice>
    <choice next="http://www.weather.example/weather.vxml">
      Weather information
    </choice>
    <choice next="#login">
      Log in
    </choice>
  </menu>

  <form id="login">
    <field name="phone_number" type="phone">
      <prompt>Please say your complete phone number</prompt>
    </field>
    <field name="pin_code" type="digits">
      <prompt>Please say your PIN code</prompt>
    </field>
    <block>
      <submit next="/servlet/login"/>
    </block>
  </form>
</vxml>
```

This document enables a dialogue like the following one in which the user selects an item from a menu:

C (computer): Say one of: Sports scores; Weather information; Log in.

H (human): Sports scores

The computer then retrieves and interprets a VoiceXML document from www.sports.example/sports.vxml containing a specification of the next segment of the dialogue (in this case, a sports information service). The dialogue uses the menu element (but not the form element). It also illustrates a capability similar to the one provided for visual applications by a set of static HTML hyperlinks. However, the static linking of information is only the Web's most basic function. The Web's most compelling feature is its dynamic distributed services, which require forms.

The form is VoiceXML's basic dialogue unit, describing a set of inputs (fields) needed from the user to complete a transaction between the user agent (browser) and a server. Each field includes a prompt and a specification of what the user is allowed to say to provide the required input. The form also specifies what to do with the set of fields after they are collected. The following dialogue between a computer and a human uses both the menu and the form in the example VoiceXML document:

C: Say one of: Sports scores; Weather information; Log in.

H: Log in.

C: Please say your complete phone number

H: 914-555-1234

C: Please say your PIN code

H: 1 2 3 4

The computer has now collected the two fields needed to complete the login, so it executes the code block containing a submit command, thus causing the information collected to be submitted to a server for processing. Each field specifies the set of acceptable user responses. Limiting these responses serves two purposes:

- Allow them to be verified and provide help (in case of an invalid response) locally without the delay of a round trip over the network to the application server; and
- Help achieve good speech-recognition accuracy, particularly over a relatively low-quality channel like a telephone.

In the example VoiceXML document, the set of acceptable user inputs is specified implicitly by specifying a "type" attribute ("phone" and "digits" in the

example) in the field element. VoiceXML interpreters provide built-in support for a set of common field types, including number, digits, phone, date, and time. However, applications would be seriously constrained if they were limited to only these built-in types. VoiceXML applications may specify their own field types using grammars, or enumerations in compact form of a set of phrases. The following example VoiceXML document illustrates the use of grammars in an online voice-enabled restaurant application:

```
<form>
  <field name="drink">
    <prompt>What would you like to drink?</prompt>
    <grammar>
      coffee | tea | orange juice | milk | nothing
    </grammar>
  </field>
  <field name="sandwich">
    <prompt>What sandwich would you like?</prompt>
    <grammar src="sandwiches.gram"/>
  </field>
  <block>
    <submit next="/servlet/order"/>
  </block>
</form>
```

The grammars here are specified using the Java Speech Grammar Format (JSGF) (see java.sun.com/products/java-media/speech/). The first one is inline and consists of a list of words and phrases ("coffee," "tea," and so on) the user may say in response to the prompt for that field. The second is contained in an external file called "sandwiches.gram":

```
<ingredient> = ham | roast beef | tomato | lettuce | swiss [cheese];
<bread> = rye | white | whole wheat;
public <sandwich> = <ingredient> {[and] <ingredient>}* on <bread>;
```

This grammar consists of three rules: The first, labeled `<ingredient>`, specifies a list of phrases listing sandwich ingredients. The last phrase ("ingredient" in this case) in the list uses square brackets to indicate the word "cheese" is optional; thus, the user may say either "swiss" or "swiss cheese." The second rule, labeled `<bread>`, specifies a list of phrases naming breads. And the third rule, labeled `<sandwich>`, specifies that a complete description of a sandwich consists of at least one ingredient, followed by zero or additional ingredients optionally separated by the word "and" and ending finally with the word "on" followed by the name of a bread. This last rule is marked "public," indicating it defines the phrases the user can actually say; the first two rules are used only in the formation of the `<sand-`

`wich>` rule. (For more on grammars, a good starting point is the JSGF reference manual at java.sun.com/products/java-media/speech/.)

A typical dialogue between a computer and a human enabled by this form might be:

C: What would you like to drink?
H: Orange juice
C: What sandwich would you like?
H: Ham, lettuce, and swiss on rye

The computer has now collected the two fields needed to complete the order, so it executes the code block containing a `submit` command, thus causing the information collected to be submitted to a server for processing.

VoiceXML and Web Standards

As the first line of the food-ordering dialogue indicates, VoiceXML is an XML application, meaning it adheres to the XML standard that at its core specifies the meta-delimiters `<`, `</`, `>`, `=`, and `"` (see www.w3.org/TR/1998/REC-xml-19980210). The rest of the contents of the example are specific either to VoiceXML ("vxml," "menu," "prompt," "choice," and "next") or to the application ("Say one of:" and "Sports scores").

Basing VoiceXML on the XML standard yields some important benefits. The most important is it allows the reuse and easy retooling of existing tools for creating, transforming, and parsing XML documents. It also allows VoiceXML to make use of other complementary XML-based standards. For example, VoiceXML applications occasionally need to specify speech-synthesis parameters, such as volume, speaking rate, and pitch. For this purpose—specifying synthesis parameters—VoiceXML incorporates the XML markup from the Java Speech Markup Language, an industry standard for speech synthesis markup (see java.sun.com/products/java-media/speech/)

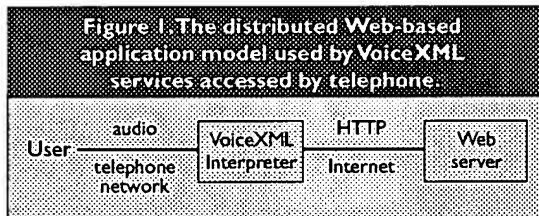
The Distributed Model

The Web brings to each user a worldwide array of information and services while bringing each information and service provider a worldwide customer base. Thus, a distributed application model is fundamental to the Web; VoiceXML builds on the same distributed model that has already proved so successful for visual Web-based services. Figure 1 outlines the distributed Web-based application model used by VoiceXML services accessed by telephone.

The VoiceXML architecture is the same as the one in the more familiar visual Web application model, except the HTML interpreter (Web browser) is

replaced by a VoiceXML interpreter, and voice replaces the mouse and keyboard as the user-interface medium. In addition to its core capabilities, VoiceXML provides more advanced features, including local validation and processing, audio playback and recording, and support for context-specific and

Finally, VoiceXML provides support for subdialogues (an entire form that is executed), the result of which is to provide an input field to another form. This feature has two uses: provide a disambiguation or confirmation dialogue for an input and support reusable subdialogues.



tapered help and for reusable subdialogues.

Local processing and validation of user input is accomplished through a collection of elements providing a more-or-less standard programming model. A "block" element allows code to be run at any point in the process of collecting inputs. A "filled" element allows input validation code to gain control upon completion of any set of user inputs; this element is particularly useful for the mixed-initiative dialogue model in which the user is able to supply inputs in any order. Finally, a "script" element allows ECMAScript (also known as JavaScript) program fragments to be run locally at any point in the dialogue (see www.ecma.ch/stand/ecma-262.htm).

The playback of prerecorded audio prompts is accomplished through an "audio" element. Recording of user messages is done through the "record" element; the recorded audio may then be played back locally using the "audio" element or uploaded to the server for storage, processing, or playback at a later time.

Meanwhile, context-specific and tapered help is provided by a built-in system of events and event handlers. VoiceXML defines a set of events corresponding to, for example, a user request for help, a failure by the user to respond within a timeout period, or user input that doesn't match an active grammar. The application may then provide (in any given context, including a form or a field) an event handler responding appropriately to a given event for a particular context. Moreover, help may be tapered; a count may be specified for each event handler so a different handler is executed, depending on how many times the event has occurred in that context. For example, tapering can be used to provide increasingly more detailed messages each time a user asks for help.

Compared with HTML

While VoiceXML reuses many concepts and designs from HTML, it differs in several ways due to the differences between visual and voice interactions. For example, an HTML document is a single unit that is fetched from a network resource specified by a uniform resource identifier and presented to the user all at once. In contrast, a VoiceXML document contains a number of dialogue units (menus or forms) presented sequentially. This difference is due to the visual medium's ability to display a number of items in parallel, while the voice medium is inherently sequential.

Thus, although a given VoiceXML document may contain the same information as a corresponding HTML document, the VoiceXML document is structured differently to reflect the sequential nature of the voice medium. So, for example, the HTML equivalent of the menu in the simple VoiceXML document outlined earlier might be:

```

Please select a service.
<a href="http://www.sports.example/sports.html">
  Sports scores
</a>
<a href="http://www.weather.example/weather.html">
  Weather information
</a>
<a href="#login">
  Log in.
</a>
  
```

In HTML, there is no need to identify this menu as a unit or to isolate it using markup structure from other elements on the same page. However, VoiceXML requires dialogue elements (menus and forms) to be identified as distinct units so they may be presented one at a time to the user. Thus, while an HTML document functions, in effect, as a single dialogue unit, a VoiceXML document is a container of dialogue units, such as menus and forms, each containing logic to sequence the interpreter to the next unit.

Another consequence of the sequential nature of the voice medium is the need for the markup to contain application logic for sequencing among dialogue units. This need is reflected in a tighter integration of sequential logic elements into VoiceXML than in

VoiceXML supports simple "directed" dialogues: the computer directs the conversation at each step by prompting the user for the next piece of information.

HTML. For example, VoiceXML contains markup elements for sequence control; in HTML, such control is available only through the relatively more cumbersome method of scripting.

Natural Dialogue

VoiceXML supports simple "directed" dialogues; the computer directs the conversation at each step by prompting the user for the next piece of information. Dialogues between humans don't operate on this simple model, of course. In a natural dialogue, each participant may take the initiative in leading the conversation. A computer-human dialogue modeled on this idea is referred to as a "mixed-initiative" dialogue, because either the computer or the human may take the initiative.

The field of spoken interfaces is not nearly as mature as the field of visual interfaces, so standardizing an approach to natural dialogue is more difficult than designing a standard language for describing visual interfaces like HTML. Nevertheless, VoiceXML takes some modest steps toward allowing applications to give users some degree of control over the conversation.

In the forms described earlier, the user was asked to supply (by speaking) a value for each field of a form in sequence. The set of phrases the user could speak in response to each field prompt was specified by a separate grammar for each field. This approach allowed the user to supply one field value in sequence. Consider a form for airline travel reservations in which the user supplies a date, a city to fly from, and a city to fly to. A directed dialogue conversation for completing such a form might proceed as follows:

C: On what date do you wish to fly?
H: February 29th.
C: From what city?
H: New York.
C: To what city?
H: Chicago.

In contrast, a somewhat more natural dialogue might proceed as follows:

C: How can I help you?
H: I'd like to fly from New York on February 29th.
C: Where would you like to fly to?
H: To Chicago.

VoiceXML enables such relatively natural dialogues by allowing input grammars to be specified at the form level, not just at the field level. A form-level grammar for these applications defines utterances that allow users to supply values for a number of fields in one utterance. For example, the utterance "I'd like to fly from New York on February 29th" supplies values for both the "from city" field and the "date" field. VoiceXML specifies a form-interpretation algorithm that then causes the browser to prompt the user for the values (one by one) of missing pieces of information (in this example, the "to city" field).

VoiceXML's special ability to accept free-form utterances is only a first step toward natural dialogue. VoiceXML will continue to evolve, incorporating more advanced features in support of natural dialogue. ■

BRUCE LUCAS (voicexml@us.ibm.com) is a software architect in the Voice Systems Group at the IBM T.J. Watson Research Center in Yorktown Heights, NY.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Rapid VXML Development! VXML 2.0 Compliant.
www.isw.com.au

- **Free VoiceXML Gateway**
Runs on PC with a voice modem. Call scheduler, toolkit. Download now!
<http://www.voicent.com>
- **Vxml**
Free **VoiceXML** guide. Find tools, tutorials, hosting services & more!
www.centipedia.com

Search for "**voicexml re-entrant**" using: [Google](#)

[Advertise](#) | [Help](#) | [Text-only Skin](#) | [Submit Site](#) | [HotBot International](#) | [Yellow Pages](#)

© Copyright 2005, Lycos, Inc. All Rights Reserved. | [Privacy Policy](#) | [Terms & Conditions](#) | [HotBot Your Site](#)



voicexml reentrant

[Homepage](#) | [Advanced Search](#)

Search using:

Ask Jeeves

Google

CUSTOM WEB FILTERS

[HotBot Skins](#) | [Prefe](#)
Date: **Before December 4 2000** [[Edit this Search](#)]

SPONSORED LINKS (filters not applied)

- **VoiceXML made darn easy:**
Voxeo **VoiceXML** hosting & platforms: Free trials, great customer service
www.voicexml-experts.com
- **XML IVR**
Automate phone calls with XML. Inbound & outbound. Try it now free
www.voiceshot.com
- **Envox 6 VoiceXML Platform**
Rapid IVR & speech app development Free evaluation software
www.envox.com
- **VoiceXML Gateway/Software**
Plum's proven **VoiceXML** IVR. Free trials available.
www.plumvoiceportals.com
- **VoiceXML Specialist**
VoiceXML services : hosting, development, training, consulting
www.eloquant.com

SI

Visibridge VoiceXML Gateway
lines Free trial incl
TTS
www.visibridge.com

Free VoiceXML
Runs on PC with a
Call scheduler, too
now!
<http://www.voi.com>

Vxml
Free **VoiceXML** g
tutorials, hosting s
www.centipedia.com

Place**Sorry, your search had no web results**

Revise your search query and try these tips:

- Check your spelling.
- Try using fewer words, or more general terms.
- Use fewer filters for your search.
- Use [Search Tips](#) to improve your search results.

Search for "**voicexml reentrant**" using: [Google](#)

[Homepage](#) | [Advanced Search](#)

Search using:

[Ask Jeeves](#)[Google](#)

CUSTOM WEB FILTERS

[HotBot Skins](#) | [Prefe](#)Date: **Before December 4 2000** [[Edit this Search](#)]

SPONSORED LINKS (filters not applied)

- **VoiceXML made darn easy:**
Voxeo **VoiceXML** hosting & platforms: Free trials, great customer service
www.voicexml-experts.com
- **Turnkey IVR & Speech**
Custom and Packaged Solutions Free - Tech./Business Assessments
www.ivr.envox.com
- **VoiceXML Gateway/Software**
Plum's proven **VoiceXML** IVR. Free trials available.
www.plumvoiceportals.com
- **VoiceXML Specialist**
VoiceXML services : hosting, development, training, consulting
www.eloquant.com
- **IVR-Xpress Visual Builder**
Rapid VXML Development! VXML 2.0 Compliant.
www.isw.com.au

SI

Free VoiceXMLRuns on PC with a
Call scheduler, too
now!<http://www.voi>**Vxml**Free **VoiceXML** g
tutorials, hosting s
www.centipedia[Place](#)**Sorry, your search had no web results**

Revise your search query and try these tips:

- Check your spelling.
- Try using fewer words, or more general terms.
- Use fewer filters for your search.
- Use [Search Tips](#) to improve your search results.

MORE SPONSORED LINKS

- **IVR-Xpress Visual Builder**